

5

Visualización de Datos de Varias Tablas

ORACLE

Objetivos

Al finalizar esta lección, debería estar capacitado para:

- **Escribir sentencias `SELECT` para acceder a datos de más de una tabla mediante las uniones igualitarias y no igualitarias**
- **Unir una tabla a sí misma mediante una autounión**
- **Ver datos que generalmente no cumplen una condición de unión utilizando uniones externas**
- **Generar un producto cartesiano de todas las filas de dos o más tablas**

Obtención de Datos de Varias Tablas

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
...		
202	Fay	20
205	Higgins	110
206	Gietz	110

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500
60	IT	1400
80	Sales	2500
90	Executive	1700
110	Accounting	1700
190	Contracting	1700



EMPLOYEE_ID	DEPARTMENT_ID	DEPARTMENT_NAME
200	10	Administration
201	20	Marketing
202	20	Marketing
...		
102	90	Executive
205	110	Accounting
206	110	Accounting

Tipos de Uniones

Las uniones compatibles con el estándar SQL:1999 son:

- **Uniones cruzadas**
- **Uniones naturales**
- **Cláusula USING**
- **Uniones externas completas (o de dos lados)**
- **Condiciones de unión arbitrarias para uniones externas**

Unión de Tablas mediante la Sintaxis SQL:1999

Utilice una unión para consultar datos de más de una tabla:

```
SELECT    table1.column, table2.column
FROM      table1
[NATURAL JOIN table2] |
[JOIN table2 USING (column_name)] |
[JOIN table2
    ON (table1.column_name = table2.column_name)] |
[LEFT|RIGHT|FULL OUTER JOIN table2
    ON (table1.column_name = table2.column_name)] |
[CROSS JOIN table2];
```


Creación de Uniones Naturales

- La cláusula **NATURAL JOIN** se basa en todas las columnas de las dos tablas que tienen el mismo nombre.
- Selecciona filas de las dos tablas que tienen valores iguales en todas las columnas correspondientes.
- Si las columnas que tienen los mismos nombres tienen tipos de datos diferentes, se devuelve un error.

Recuperación de Registros con Uniones Naturales

```
SELECT department_id, department_name,  
       location_id, city  
FROM   departments  
NATURAL JOIN locations ;
```

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
60	IT	1400	Southlake
50	Shipping	1500	South San Francisco
10	Administration	1700	Seattle
90	Executive	1700	Seattle
110	Accounting	1700	Seattle
190	Contracting	1700	Seattle
20	Marketing	1800	Toronto
80	Sales	2500	Oxford

8 rows selected.

Creación de Uniones con la Cláusula USING

- Si hay varias columnas que tienen los mismos nombres pero los tipos de datos no se corresponden, la cláusula **NATURAL JOIN** se puede modificar mediante la cláusula **USING** para especificar las columnas que se deben utilizar para una unión igualitaria.
- Utilice la cláusula **USING** para asignar sólo una columna cuando corresponde a más de una columna.
- No utilice un alias o un nombre de tabla en las columnas a las que se hace referencia.
- Las cláusulas **NATURAL JOIN** y **USING** se excluyen mutuamente.

Unión de Nombres de Columna

EMPLOYEES

EMPLOYEE_ID	DEPARTMENT_ID
200	10
201	20
202	20
124	50
141	50
142	50
143	50
144	50
103	60
104	60
107	60
149	80
174	80
176	80

...

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
20	Marketing
50	Shipping
50	Shipping
50	Shipping
50	Shipping
50	Shipping
50	Shipping
60	IT
60	IT
60	IT
80	Sales
80	Sales
80	Sales

...

↑
Clave ajena

↑
Clave primaria

ORACLE

Recuperación de Registros con la Cláusula USING

```
SELECT employees.employee_id, employees.last_name,  
       departments.location_id, department_id  
FROM   employees JOIN departments  
USING (department_id) ;
```

EMPLOYEE_ID	LAST_NAME	LOCATION_ID	DEPARTMENT_ID
200	Whalen	1700	10
201	Hartstein	1800	20
202	Fay	1800	20
124	Mourgos	1500	50
141	Rajs	1500	50
142	Davies	1500	50
144	Vargas	1500	50
143	Matos	1500	50

...

19 rows selected.

Cualificación de Nombres de Columna Ambiguos

- **Utilice prefijos de tabla para cualificar nombres de columna que estén en varias tablas.**
- **Utilice prefijos de tabla para mejorar el rendimiento.**
- **Utilice alias de columna para distinguir columnas que tengan nombres idénticos pero que residan en tablas diferentes.**
- **No utilice alias en columnas que estén identificadas en la cláusula `USING` y que se muestren en cualquier otra parte de la sentencia SQL.**

Uso de Alias de Tabla

- Utilice alias de tabla para simplificar las consultas.
- Utilice alias de tabla para mejorar el rendimiento.

```
SELECT e.employee_id, e.last_name,  
       d.location_id, department_id  
FROM   employees e JOIN departments d  
USING (department_id);
```


Creación de Uniones con la Cláusula ON

- La condición de unión para la unión natural es básicamente una unión igualitaria de todas las columnas con el mismo nombre.
- Utilice la cláusula ON para especificar condiciones arbitrarias o para especificar las columnas que se unirán.
- La condición de unión se separa de otras condiciones de búsqueda.
- La cláusula ON facilita la comprensión del código.

Recuperación de Registros con la Cláusula ON

```
SELECT e.employee_id, e.last_name, e.department_id,  
       d.department_id, d.location_id  
FROM   employees e JOIN departments d  
ON     (e.department_id = d.department_id);
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
200	Whalen	10	10	1700
201	Hartstein	20	20	1800
202	Fay	20	20	1800
124	Mourgos	50	50	1500
141	Rajs	50	50	1500
142	Davies	50	50	1500
143	Matos	50	50	1500

...

19 rows selected.

ORACLE

Autouniones mediante la Cláusula ON

EMPLOYEES (WORKER)

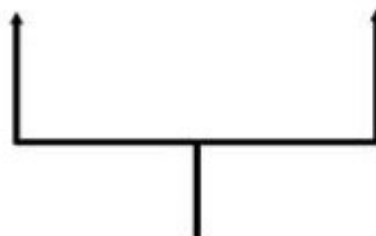
EMPLOYEE_ID	LAST_NAME	MANAGER_ID
100	King	
101	Kochhar	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100

...

EMPLOYEES (MANAGER)

EMPLOYEE_ID	LAST_NAME
100	King
101	Kochhar
102	De Haan
103	Hunold
104	Ernst
107	Lorentz
124	Mourgos

...



**MANAGER_ID en la tabla WORKER es igual que
EMPLOYEE_ID en la tabla MANAGER.**

Autouniones mediante la Cláusula ON

```
SELECT e.last_name emp, m.last_name mgr
FROM   employees e JOIN employees m
ON     (e.manager_id = m.employee_id);
```

EMP	MGR
Hartstein	King
Zlotkey	King
Mourgos	King
De Haan	King
Kochhar	King

...

19 rows selected.

Aplicación de Condiciones Adicionales a una Unión

```
SELECT e.employee_id, e.last_name, e.department_id,  
       d.department_id, d.location_id  
FROM   employees e JOIN departments d  
ON      (e.department_id = d.department_id)  
AND     e.manager_id = 149 ;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
174	Abel	80	80	2500
176	Taylor	80	80	2500

Creación de Uniones en Tres Sentidos con la Cláusula ON

```
SELECT employee_id, city, department_name
FROM   employees e
JOIN   departments d
ON     d.department_id = e.department_id
JOIN   locations l
ON     d.location_id = l.location_id;
```

EMPLOYEE_ID	CITY	DEPARTMENT_NAME
103	Southlake	IT
104	Southlake	IT
107	Southlake	IT
124	South San Francisco	Shipping
141	South San Francisco	Shipping
142	South San Francisco	Shipping
143	South San Francisco	Shipping
144	South San Francisco	Shipping

19 rows selected.

Uniones No Igualitarias

EMPLOYEES

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Hunold	9000
Ernst	6000
Lorentz	4200
Mourgos	5800
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500
Zlotkey	10500
Abel	11000
Taylor	8600

...

20 rows selected.

JOB_GRADES

GRA	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

El salario de la tabla
EMPLOYEES debe estar
entre el salario más bajo
y el más alto de la tabla
JOB_GRADES.

ORACLE

Recuperación de Registros con Uniones No Igualitarias

```
SELECT e.last_name, e.salary, j.grade_level
FROM employees e JOIN job_grades j
ON e.salary
   BETWEEN j.lowest_sal AND j.highest_sal;
```

LAST_NAME	SALARY	GRA
Matos	2600	A
Vargas	2500	A
Lorentz	4200	B
Mourgos	5800	B
Rajs	3500	B
Davies	3100	B
Whalen	4400	B
Hunold	9000	C
Ernst	6000	C

20 rows selected.

Uniones Externas

DEPARTMENTS

DEPARTMENT_NAME	DEPARTMENT_ID
Administration	10
Marketing	20
Shipping	50
IT	60
Sales	80
Executive	90
Accounting	110
Contracting	190

8 rows selected.

EMPLOYEES

DEPARTMENT_ID	LAST_NAME
90	King
90	Kochhar
90	De Haan
60	Hunold
60	Ernst
60	Lorentz
50	Mourgos
50	Rajs
50	Davies
50	Matos
50	Vargas
80	Zlotkey

...

20 rows selected.

No hay empleados en el departamento 190.

ORACLE

Uniones INNER frente a OUTER

- En SQL:1999, la unión de dos tablas que devuelve sólo filas con correspondencia se denomina unión interna.
- Una unión entre dos tablas que devuelve los resultados de la unión interna y las filas sin correspondencia de las tablas a la izquierda (o derecha) se denomina unión externa izquierda (o derecha).
- Una unión entre dos tablas que devuelve los resultados de la unión interna y los resultados de una unión izquierda y derecha es una unión externa completa.

LEFT OUTER JOIN

```
SELECT e.last_name, e.department_id, d.department_name  
FROM   employees e LEFT OUTER JOIN departments d  
ON     (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
...		
De Haan	90	Executive
Kochhar	90	Executive
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		

20 rows selected.

RIGHT OUTER JOIN

```
SELECT e.last_name, e.department_id, d.department_name
FROM   employees e RIGHT OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
Davies	50	Shipping
...		
Kochhar	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
	190	Contracting

20 rows selected.

FULL OUTER JOIN

```
SELECT e.last_name, d.department_id, d.department_name
FROM   employees e FULL OUTER JOIN departments d
ON     (e.department_id = d.department_id) ;
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
...		
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		
	190	Contracting

21 rows selected.

Productos Cartesianos

- **Se forma un producto cartesiano cuando:**
 - Se omite una condición de unión
 - Una condición de unión no es válida
 - Todas las filas de la primera tabla se unen a todas las filas de la segunda tabla
- **Para evitar un producto cartesiano, incluya siempre una condición de unión válida.**

Generación de Productos Cartesianos

EMPLOYEES (20 filas)

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
...		
202	Fay	20
205	Higgins	110
206	Gietz	110

20 rows selected.

DEPARTMENTS (8 filas)

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500
60	IT	1400
80	Sales	2500
90	Executive	1700
110	Accounting	1700
190	Contracting	1700

8 rows selected.

Producto cartesiano:
20 x 8 = 160 filas

EMPLOYEE_ID	DEPARTMENT_ID	LOCATION_ID
100	90	1700
101	90	1700
102	90	1700
103	60	1700
104	60	1700
107	60	1700

...

160 rows selected.

ORACLE

Creación de Uniones Cruzadas

- La cláusula **CROSS JOIN** genera el producto combinado de dos tablas.
- También se denomina un producto cartesiano entre las dos tablas.

```
SELECT last_name, department_name  
FROM employees  
CROSS JOIN departments ;
```

LAST_NAME	DEPARTMENT_NAME
King	Administration
Kochhar	Administration
De Haan	Administration
Hunold	Administration

...

160 rows selected.

Resumen

En esta lección, ha aprendido a utilizar uniones para mostrar datos de varias tablas mediante:

- **Uniones igualitarias**
- **Uniones no igualitarias**
- **Uniones externas**
- **Autouniones**
- **Uniones cruzadas**
- **Uniones naturales**
- **Uniones externas completas (o de dos lados)**

Práctica 5: Visión General

Esta práctica cubre los temas siguientes:

- **Unión de tablas mediante una unión igualitaria**
- **Realización de uniones externas y autouniones**
- **Agregación de condiciones**